

## **BAB II**

### **KAJIAN TEORI**

#### **A. Pengertian Sistem**

Pada dasarnya, dalam kehidupan sehari-hari sistem merupakan suatu cara, misalnya sistem pengamatan, sistem penilaian, sistem pengajaran dan lain-lain. Berikut beberapa pengertian sistem, yaitu

1. Sistem merupakan jaringan elemen-elemen (*software, hardware* dan *brainware*) yang saling berhubungan, membentuk satu kesatuan untuk melaksanakan suatu tujuan pokok yaitu mengolah suatu data menjadi informasi. (Jogiyanto, 1999)
2. Sistem adalah sekumpulan objek atau elemen yang dipandang sebagai keseluruhan dan dirancang untuk mencapai suatu tujuan dari sistem tersebut. (Hariyanto, 1993)

#### **B. Pengertian Data dan *File***

Untuk menuju pada pengertian sistem penyimpanan terdistribusi, diperlukan pemahaman yang tentang data dan *file*. Berikut beberapa pengertian data menurut para ahli, yaitu

1. Data merupakan kumpulan kejadian yang diangkat dari suatu kenyataan, dapat berupa angka, huruf, simbol atau gabungan dari ketiganya (Jogiyanto, 1999).
2. Data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, vektor dan sebagainya. (Teguh Wahyono, 2004)

Suatu data dapat memuat hal yang penting dan rahasia. Untuk itu data yang bersifat penting dan rahasia tentunya disimpan di tempat yang aman, contohnya data disimpan dalam bentuk *file* dalam komputer.

Berikut beberapa pengertian tentang *file* dari ahli, di antaranya adalah

1. *File* adalah kumpulan dari *record-record* yang berhubungan dalam satu kesatuan. (Jogiyanto, 1999)
2. *File* merupakan sebuah blok informasi yang terbentuk dari beberapa *byte* disimpan secara bersamaan dalam sebuah media penyimpanan dalam komputer. (Nani Mintarsih, 2012)
3. *File* adalah unit penyimpanan logika yang berisi kumpulan informasi yang berhubungan. (Fajrillah, 2011)

Adapun operasi-operasi pokok yang dilakukan untuk suatu *file* diantaranya adalah membuat *file*, menulis *file*, membaca *file*, reposisi dalam *file*, menghapus *file* dan memotong *file*. Adapun operasi tambahan dalam sebuah *file* adalah menambah, mengubah nama dan membuat duplikasi *file*.

### **C. Sistem Penyimpanan Terdistribusi**

Setelah mengkaji pengertian sistem, data dan *file*, selanjutnya akan dibahas tentang sistem penyimpanan terdistribusi, diawali dengan pengertian sistem terdistribusi dahulu. Menurut Silberschatz, dkk. (2012) bahwa sistem terdistribusi adalah kumpulan *node* (titik) yang digabungkan dengan elemen-elemen seperti *hardwere*, *software* dan dijalankan *brainwere* dan saling terhubung oleh jaringan komunikasi. Definisi lain dikemukakan Coulories, dkk. (2012) bahwa sistem terdistribusi adalah salah satu komponen yang

terletak di jaringan komputer yang dapat berkomunikasi dan mengkoordinasi antar elemen atau objek komputer dengan proses distribusi melewati pesan.

Berikut beberapa keuntungan dari sistem terdistribusi, yaitu:

1. *Resource sharing*
2. Komunikasi

Dewasa ini, sistem penyimpanan data mengalami perkembangan yang semakin canggih dan modern yang didasarkan atas sistem terdistribusi. Penyimpanan data ini dapat disimpan pada media penyimpanan, seperti *disket*, *compact disk*, *flashdisk*, *harddisk* dan lain-lain, bahkan menyimpan suatu data dapat dilakukan pada media *online*, seperti pada *email* dan *dropbox*. Penyimpanan pada media *online* ini adalah salah satu penerapan sistem penyimpanan terdistribusi (*Distributed storage system*). Berikut beberapa pengertian sistem penyimpanan terdistribusi, yaitu:

1. Menurut Tanakorn Chareonvisal (2012) dalam thesisnya, sistem penyimpanan terdistribusi adalah sistem penyimpanan data yang mendistribusikan data ke beberapa titik penyimpanan.
2. Sistem penyimpanan terdistribusi adalah suatu cara penyimpanan yang dapat mendistribusikan data di beberapa disk yang telah secara alami diperluas ke beberapa *node* penyimpanan yang saling berhubungan melalui jaringan atau dengan kata lain sistem penyimpanan terdistribusi adalah jaringan sistem penyimpanan yang didistribusikan yang mengacu pada sistem penyimpanan data secara terdistribusi di seluruh jaringan

komputer, di mana interkoneksi memainkan peran penting (Oggier Frederique dan Datta Anwitaman, 2012).

#### **D. Network Coding**

*Network coding* adalah sebuah skema baru di dalam teori informasi yang memungkinkan untuk melakukan partisi *file* dan penggabungan *file* sebelum diteruskan dengan adanya proses enkripsi atau *encoding* dan dekripsi atau *decoding*. *Enkripsi atau encoding* adalah suatu cara atau metode dalam teori pengkodean yang mengubah suatu data asli menjadi kode-kode yang melambangkan data tersebut, sedangkan dekripsi atau *decoding* adalah kebalikan dari enkripsi *encoding*, yaitu suatu cara atau metode dalam teori pengkodean yang mengubah kode-kode data tersebut menjadi data asli. (Rinaldi Munir, 2006)

*Network coding* digunakan untuk mempercepat akses distribusi data pada jaringan komputer. Kinerja *network coding* lebih cepat dan efisien karena data dikirim berbentuk kode secara kombinasi dari beberapa *source* atau sumber dengan satu kali pengiriman data. Pada dasarnya kinerja *network coding* adalah mempartisi sebuah *file* menjadi beberapa bagian dan data tersebut dikirim secara bersama dan setelahnya partisi data diterima dengan komplit, selanjutnya dari beberapa kombinasi kode data tersebut digabungkan dan diterjemahkan sehingga menjadi data asli (Kashif Mahmood, 2010). Kombinasi dari partisi kode tersebut dalam pengkodean jaringan ini disebut *network coding* (Tanakorn Chareonvisal, 2012). Berikut merupakan beberapa keuntungan *networking coding*, yaitu: (Rickard Browman, 2013)

1. Meningkatkan throughput (perhitungan nilai pesan yang sukses dikirim)
2. Menambah keamanan

#### E. Skema *Redundancy*

Sistem penyimpanan terdistribusi menyediakan akses yang dapat diandalkan untuk data melalui *node-node* penyimpanan tersebar yang telah dibuat. Sistem penyimpanan terdistribusi membutuhkan *redundancy* untuk melakukan *repair* suatu kegagalan data dengan membuat *node* penyimpanan tambahan sebagai tempat *backup* suatu *file*. Untuk meningkatkan *reliability* dalam komunikasi perlu ditambah *redundancy* yang dapat digunakan sebagai membenarkan kesalahan (Scot and Paul, 1989), misalnya:

A dikodekan dengan 00

C dikodekan dengan 01

B dikodekan dengan 10

D dikodekan dengan 11

Jika pesan yang dikirim harusnya kata ADA dengan kode 00 11 00 tetapi terjadi kesalahan pengiriman pesan, setelah itu dilakukan redundant atau penambahan digit untuk membenarkan kesalahan

A dikodekan dengan 00000

B dikodekan dengan 01010

C dikodekan dengan 10001

D dikodekan dengan 11100

Andai pesan yang dikirim berkode 00000 10100 00000. Selanjutnya dianalisis antara perbedaan kode pesan yang salah kirim yaitu 10100 dibandingkan dengan pengkodean karakter yang ada dan dicari perbedaan kode yang paling sedikit perbedaannya.

10100 dengan A = 2 perbedaan kode

10100 dengan B = 2 perbedaan kode

10100 dengan C = 4 perbedaan kode

10100 dengan D = 1 perbedaan kode

Setelah dilakukan analisis dipilih karakter pesan dengan perbedaan kode yang sedikit yaitu D, jadi pesan yang diharapkan adalah kata ADA.

Skema *redundancy* dapat menggunakan kode replikasi (*repetition code* atau *replication code*) dan kode penghapusan (*erasure code*). Berikut sedikit penjelasan dari kode replikasi (*repetition code* atau *replication code*) dan kode penghapusan (*erasure code*).

### **1. Repetition code atau Replication code**

Metode *replication code* adalah metode mereplika atau menggandakan suatu kode sebagai *backup* data. Cara kerja metode *replication code* adalah dengan membagi *file* berukuran *M* menjadi *k* bagian penyimpanan, yang masing-masing titik penyimpanan menyimpan *M/k* fragmen atau dengan kata lain teknik untuk melakukan *copy paste* data ke dalam titik penyimpanan yang tersebar melalui jaringan.

Sistem kerja dari *replication code* dalam melakukan perbaikan data adalah ketika terjadi kegagalan pada sebuah titik penyimpanan, maka terdapat titik penyimpanan lain yang masih hidup yang isinya sama dengan partisi data aslinya. Titik penyimpanan *backup file* terhubung ke sebuah *node* penyimpanan data yang gagal dan dapat memperbaikinya sesuai isi titik penyimpanan yang gagal tersebut. (Tanakorn Chareonvisal, 2012)

## 2. *Erasure code*

*Erasure code* adalah metode yang hampir sama dengan replikasi yang membagi objek data awal atau asli ke dalam  $k$  bagian penyimpanan yang selanjutnya digunakan untuk membangun  $n$  sandi bagian (Dimakis, dkk, 2007). Dalam skema *redundancy erasure code* ada dua kelas metode yang digunakan dalam sistem penyimpanan terdistribusi, yaitu:

### a) **MDS (*Maximum Distance Separable*) Code**

Metode *MDS* merupakan salah satu kode penghapusan yang maksimal untuk mengoreksi kesalahan. Kode ini meningkatkan penyimpanan secara efisien, tetapi membutuhkan kompleksitas yang lebih tinggi dari kode replikasi. *File* asli dibagi ke dalam  $M$  fragmen yang disimpan dalam  $n$  titik penyimpanan. Masing-masing titik penyimpanan menyimpan  $M/k$  data fragmen dan  $k$  titik penyimpanan dibutuhkan untuk membuat kembali data aslinya. (Rickard Browman, 2013)

### b) ***Regenerating Code***

Kode regenerasi merupakan variasi lain dari kode penghapusan, di mana kode regenerasi meningkatkan fungsi perbaikan karena faktanya kode regenerasi memperbolehkan titik penyimpanan baru untuk menghubungkan semua titik penyimpanan yang ada jika terjadi kesalahan titik. Artinya  $k \leq d \leq n-1$  untuk *regenerating code* dibandingkan  $d = k$  untuk *MDS code*. (Rickard Browman, 2013)

## F. PHP (*Hypertext Preeprocesor*)

*PHP (Hypertext Preeprocesor)* merupakan bahasa program standar yang berbentuk script yang dapat berjalan pada *server* (dunia *website*) yang hasilnya ditampilkan pada klien. Pada awalnya *PHP* diciptakan oleh Rasmus Lerdorf sekitar tahun 1994 yang membuat sebuah *script perl*. Kemudian sedikit demi sedikit programmer menyukai script ini dan selanjutnya dibentangkan sebagai *package*, yaitu *Personal Home Page Tool*. *Personal Home Page Tool* merupakan *engine* baru untuk *script* ini dan menciptakan tool yang lain untuk mengambil input dari *HTML* form: *FI (Form Interpreter)*. *Form Interpreter* juga dikenal dengan *PHP 2* yang berlaku sekitar tahun 1995.

Para pengguna *Form Interpreter* menggunakan tool ini ke dalam pembangunan *script* yang lebih rumit lagi sehingga berkembang *PHP 3* dan selanjutnya berkembang *PHP 4* yang aplikasi pertamanya terdiri dari *Zend engine*. *Zend engine* ini juga termasuk ke dalam *package* seperti *MySQL* untuk meningkatkan mutu simpanan prosedur di dalam *database*. *PHP* versi 4 adalah bahasa *scripting* yang menyatu dengan *HTML* dan berada di *server* sehingga perintah yang kita buat akan dijalankan dan dikerjakan di *server* dan disertai halaman *HTML* biasa. *PHP* ini bertujuan untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Hampir seluruh aplikasi berbasis web dapat dibuat dengan *PHP*, namun fungsi *PHP* yang paling utama adalah menghubungkan *database* dengan web.

Adapun kriteria yang harus diperhatikan dalam penulisan *script PHP* adalah sebagai berikut: (Bunafit Nugroho, 2004)



- a. Setiap halaman yang mengandung script *PHP* harus disimpan menggunakan ekstensi *PHP*.
- b. Setiap *script PHP* harus didahului dengan pembuka *PHP* ( `<? php`) dan diakhiri dengan penutup ( `?>`)
- c. Setiap akhir baris perintah harus di akhiri dengan titik koma ( ; )
- d. Semua variable harus diberi tanda string dolar ( \$ )
- e. Penulisan keterangan didahului dengan pembuka (`/*`) atau (`//`) dan diakhiri dengan penutup (`*/`) atau (`//`)
- f. Semua *script HTML* yang akan digabungkan dengan *script PHP* harus dihilangkan tanda petiknya ( “ ”).

Berikut beberapa kelebihan bahasa pemrograman *PHP* dari bahasa pemrograman yang lain, yaitu: (Sutarman, 2003)

- a. Bisa membuat *Web* menjadi dinamis.
- b. *PHP* bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
- c. Program yang dibuat dengan *PHP* bisa dijalankan oleh semua sistem operasi karena *PHP* berjalan secara *Web Base* yang artinya semua sistem operasi bahkan *HP* yang mempunyai *Web Browser* dapat menggunakan program *PHP*.
- d. Aplikasi *PHP* lebih cepat dibandingkan dengan *ASP* maupun *Java*.
- e. Mendukung paket *database* seperti *MySQL*, *Oracle* dan *PostgrSQL*
- f. Banyak *Web Server* yang mendukung *PHP*

- g. Pengembangan aplikasi *PHP* mudah karena banyak dokumentasi, refrensi & *developer* yang membantu dalam pengembangannya.
- h. Banyak aplikasi dan program *PHP* yang gratis dan siap pakai seperti *WordPress* dan *PrestaShop*.

## **G. Keterbagian dan Kekongruenan**

### **1. Keterbagian**

Dalam sistem penyimpanan terdistribusi identik pada partisi data atau *file* sehingga partisi-partisi *file* tersebut dapat didistribusikan ke beberapa *node* penyimpanan dalam komputer dengan teori pengkodean.

Mempartisi *file* sama halnya dengan membagi *file*. Mempartisi atau membagi *file* menjadi beberapa bagian memerlukan ilmu matematika dalam menerapkannya dikarenakan jika membagi atau mempartisi *file* tersebut tidak sesuai maka *file* tersebut jika digabungkan akan mengalami *error* atau *file* tidak utuh kembali, untuk itu diperlukan materi keterbagian, algoritma pembagian dan materi kekongruenan dalam aljabar khususnya dalam teori bilangan. Semesta pembicaraan dalam teori bilangan adalah himpunan bilangan bulat yang dinyatakan dalam huruf latin kecil ( $a, b, c, \dots, m, n$ , dan sebagainya) baik bernilai positif, negatif atau nol.

#### **Definisi 2.1 (Sukirman, 2006)**

Bilangan bulat  $a$  membagi (habis) bilangan bulat  $n$  ( $a \mid b$ ) jika dan hanya jika ada bilangan bulat  $k$  sedemikian hingga  $b = ka$ . Jika  $a$  tidak membagi habis  $b$  maka ditulis  $a \nmid b$ .

### Contoh 2.1

$3 \mid 15$ , karena ada bilangan bulat  $k = 5$ , sedemikian hingga  $3 \cdot 5 = 15$

$4 \mid -12$ , karena ada bilangan bulat  $k = -3$ , sedemikian hingga  $4 \cdot (-3) = -12$

Secara umum, jika  $a$  adalah suatu bilangan bulat dan  $b$  suatu bilangan bulat positif, maka ada tepat satu bilangan bulat  $p$  dan  $q$  sedemikian sehingga  $a = pb + q$ ,  $0 \leq q < b$ . Bilangan bulat  $p$  disebut sebagai hasil bagi dan  $q$  disebut sebagai sisa pembagian. Jika  $q = 0$  maka dikatakan  $a$  habis dibagi oleh  $b$  dan ditulis  $b \mid a$ . Jika  $q \neq 0$  maka ditulis  $b \nmid a$ .

Sifat-sifat keterbagian:

- a.  $a \mid a$  ( sifat refleksif)
- b.  $a \mid b$  dan  $b \mid c$  maka  $a \mid c$  ( sifat transitif)
- c.  $a \mid b$  maka  $a \mid mb$  , untuk setiap bilangan bulat  $m$ .
- d.  $a \mid b$  dan  $a \mid c$  maka  $a \mid b + c$  ,  $a \mid b - c$  atau  $a \mid bc$
- e.  $ab \mid c$  maka  $b \mid c$  dan  $a \mid c$
- f.  $a \mid b$  dan  $a \mid c$  maka  $a \mid ( mb + nc )$  untuk setiap bilangan bulat  $m$  dan  $n$ .

Bukti:

a.  $a1 = 1a = a$

$a \mid ka$  dengan  $k$  adalah bilangan bulat,  $k = 1$

$$a \mid 1a = a \mid a1 \text{ sehingga } a \mid a$$

b.  $a \mid b$  maka  $b = ka$  dan  $b \mid c$  maka  $c = mb$

$$c = mb = m(ka) = (mk) a.$$

Jadi  $a \mid c$ .

c. karena  $a \mid b$  maka ada bilangan bulat  $k$  sehingga  $b = ka$

jika kedua ruas dikalikan dengan  $m$  maka diperoleh

$$mb = (km)a \text{ ini berarti } a \mid mb$$

d. dengan menggunakan sifat ke 3

$a \mid mb$  dan  $a \mid mc$  dengan  $m$  bilangan bulat maka diperoleh

$$a \mid m(b + c) \text{ ini berarti } a \mid b + c$$

$$a \mid m(b - c) \text{ ini berarti } a \mid b - c$$

$$a \mid (mb)(mc) \text{ ini berarti } a \mid m^2bc = a \mid bc$$

e.  $ab \mid c \Rightarrow c = (ab)k = a(bk) \Rightarrow a \mid c$

$$ab \mid c \Rightarrow c = (ab)k = (ba)k = b(ak) \Rightarrow a \mid c$$

f. dengan sifat 3

$$a \mid b = a \mid mb, \text{ untuk setiap } m \text{ bilangan bulat}$$

$$a \mid c = a \mid nc, \text{ untuk setiap } n \text{ bilangan bulat,}$$

dan

dengan menggunakan sifat 4

$$a \mid mb + nc, \text{ untuk setiap } m \text{ dan } n \text{ bilangan bulat}$$

### **Definisi 2.2 (Sukirman, 2006)**

Jika  $a$  dan  $b$  adalah bilangan bulat, maka bilangan bulat  $d$  disebut faktor persekutuan dari  $a$  dan  $b$  jika  $d \mid a$  dan  $d \mid b$

### **Definisi 2.3 (Sukirman, 2006)**

Jika  $a$  dan  $b$  adalah bilangan bulat yang sekurang – kurangnya satu di antaranya tidak sama dengan nol, maka bilangan bulat  $d$  disebut faktor persekutuan terbesar dari  $a$  dan  $b$  diberi simbol “ $(a,b)$ ” adalah suatu bilangan bulat positif, misalnya  $d$ , yang memenuhi

$d \mid a$  dan  $d \mid b$ , serta jika  $e \mid a$  dan  $e \mid b$ , maka  $e \leq d$

**Teorema 2.1 (Sukirman, 2006)**

Jika  $a$  dan  $b$  bilangan - bilangan bulat dengan  $b > 0$ , maka ada dengan tunggal pasangan bilangan-bilangan bulat  $q$  dan  $r$  yang memenuhi  $a = qb + r$ , dengan  $0 \leq r < b$ . (ALGORITMA PEMBAGIAN)

Bukti:

Dibentuk himpunan  $S = \{ a - xb \mid x \text{ bilangan bulat dan } a - xb \geq 0 \}$

$S$  bukan himpunan kosong, sebab jika  $x = -|a|$  dan karena  $b > 0$ , maka  $(a - xb)$  anggota  $S$ , sehingga  $S$  merupakan himpunan dari bilangan bulat tak negatif maka memiliki anggota terkecil misal  $r$ . Sesuai dengan bentuk anggota  $S$ , maka  $r = a - bq$ , untuk suatu bilangan bulat  $q$  dan  $r \geq 0$ . Selanjutnya akan ditunjukkan bahwa  $r < b$

Andai  $r \geq b$  maka  $r = b + k$  dengan  $k \geq 0$ , jadi  $k = r - b$ , karena  $r = a - qb$ , maka  $k = a - qb - b = a - (q + 1)b$ . Ini berarti  $k$  merupakan anggota  $S$ .

Tetapi  $0 \leq k = r - b < r$ , hal ini tidak mungkin, karena  $r$  adalah bilangan bulat tak negatif yang terkecil dalam  $S$ , oleh karena itu pengandaian tersebut harus di negasikan. Jadi  $r < b$ , sehingga ada  $q$  dan  $r$  sedemikian sehingga  $a = qb + r$  dengan  $0 \leq r < b$ . Selanjutnya akan ditunjukan ketunggalan dari  $q$  dan  $r$ , Misal  $a$  mempunyai dua representasi yaitu  $a = qb + r = q'b + r'$  dengan  $0 \leq r < b$  dan  $0 \leq r' < b$ ,

maka  $r - r' = b(q - q')$

$|r - r'| = b|q - q'|$ , karena  $a > 0$

Dari  $-b < -r < 0$  dan  $0 \leq r' < b$

diperoleh  $-b < r' - r < b$  atau  $|r' - r| < b$

Jadi  $b |q' - q| < b$ , yang menghasilkan  $0 \leq |q' - q| < 1$

Karena  $|q' - q|$  adalah bilangan bulat tak negatif, maka hanya mungkin jika  $|q' - q| = 0$ , yaitu  $q' = q$ , sehingga  $r' = r$  juga.

Setelah membahas algoritma pembagian seperti di atas selanjutnya akan dibahas lebih mendalam lagi dari keterbagian dalam himpunan bilangan bulat yang disebut kekongruenan.

## 2. Kekongruenan

Setelah menelaah algoritma pembagian, selanjutnya akan ditelaah lebih dalam mengenai kekongruenan yang didasarkan atas keterbagian dan algoritma pembagian.

### Definisi 2.4 (Sukirman, 2006)

Jika  $m$  suatu bilangan positif, maka  $a$  kongruen dengan  $b$  modulo  $m$  dapat dinotasikan  $a \equiv b \pmod{m}$  jika  $m$  membagi  $(b - a)$ . Jika  $m$  tidak membagi  $(a - b)$  maka  $a$  tidak kongruen dengan  $b$  modulo  $m$  dinotasikan dengan  $a \not\equiv b \pmod{m}$ .

### Contoh 2.4

$16 \equiv 1 \pmod{5}$ , karena  $(16 - 1) = 15$ , 15 terbagi oleh 5

$23 \equiv 2 \pmod{7}$ , karena  $(23 - 2) = 21$ , 21 terbagi oleh 7

$43 \equiv 3 \pmod{8}$ , karena  $(43 - 3) = 40$ , 40 terbagi oleh 8

### Teorema 2.2 (Sukirman, 2006)

$a \equiv b \pmod{m}$  jika dan hanya jika ada bilangan bulat  $k$  sehingga dapat dinyatakan  $a = mk + b$

bukti:

dari definisi 2.4, jika  $m > 0$  maka  $m \mid (a - b)$  jika dan hanya jika  $a \equiv b \pmod{m}$ . Jika  $m \mid (a - b)$  maka ada bilangan bulat  $k$  sehingga  $(a - b) = mk$  untuk setiap bilangan bulat  $k$ . Karena  $a - b = mk$  artinya  $a = mk + b$ , maka  $a \equiv b \pmod{m}$  jika dan hanya jika  $a = mk + b$

### **Contoh 2.5**

$19 \equiv 1 \pmod{3}$  artinya sama dengan  $18 = 3 \cdot 6 + 1$

$23 \equiv 3 \pmod{4}$  artinya sama dengan  $23 = 4 \cdot 5 + 3$

### **Teorema 2.3 (Sukirman, 2006)**

Setiap bilangan bulat kongruen modulo  $m$  dengan tepat satu di antara  $0, 1, 2, \dots, (m-1)$

bukti:

jika  $a$  dan  $m$  bilangan bulat dan  $m > 0$ , menurut algoritma pembagian, maka  $a$  dapat dinyatakan sebagai  $a = mq + r$  dengan  $0 \leq r < m$  artinya  $a - r = mq$  yaitu  $a \equiv r \pmod{m}$ , karena  $0 \leq r < m$ , maka ada  $m$  buah pilihan untuk  $r$ , yaitu  $0, 1, 2, \dots, (m-1)$ . Jadi setiap bilangan bulat akan kongruen modulo  $m$  dengan tepat satu di antara  $0, 1, 2, \dots, (m-1)$ .

### **Teorema 2.4 (Sukirman, 2006)**

$a \equiv b \pmod{m}$  jika dan hanya jika  $a$  dan  $b$  memiliki sisa yang sama jika dibagi  $m$ .

bukti:

karena  $a \equiv b \pmod{m}$  ada bilangan bulat  $r$  dengan  $r$  adalah residu terkecil modulo  $m$   $0 \leq r < m$  maka  $a \equiv r \pmod{m}$  dan  $b \equiv r \pmod{m}$

$a \equiv r \pmod{m}$  berarti  $a = mp + r$  untuk bilangan bulat  $p$

$b \equiv r \pmod{m}$  berarti  $b = mq + r$  untuk bilangan bulat  $q$

jadi  $a$  dan  $b$  memiliki sisa yang sama jika dibagi  $m$ , dari dua persamaan di

atas diperoleh  $a - b = m(p - q)$  berarti  $m \mid (a - b)$  atau  $a \equiv b \pmod{m}$

## H. Kombinsai Linear dan Bebas Linear

### 1. Ruang Vektor

Sebelum menginjak ke teori tentang kombinasi linear dan bebas linear akan dibahas terlebih dahulu tentang ruang vektor.

#### Definisi 2.8 (Howard Anton, 1987)

Misalkan  $V$  sebarang himpunan yang dua operasinya didefinisikan yaitu penjumlahan dan perkalian dengan skalar. Penjumlahan tersebut untuk mengasosiasikan sebuah aturan dengan setiap pasang  $u$  dan  $v$  pada  $V$ , yang mengandung elemen  $u + v$ , yang dinamakan jumlah  $u$  dan  $v$ . Perkalian skalar diartikan aturan untuk mengasosiasikan skalar  $k$  maupun setiap  $u$  pada  $V$  yang mengandung elemen  $ku$ , yang dinamakan perkalian skalar  $u$  oleh  $k$ . Jika semua aksioma berikut terpenuhi oleh semua  $u, v, w$  pada  $V$  oleh semua skalar  $k$  dan  $l$ , maka  $V$  adalah ruang vektor dan anggota pada  $V$  dinamakan vektor.

- a. Jika  $u$  dan  $v$  adalah vektor pada  $V$  maka  $u + v$  berada pada  $V$
- b.  $u + v = v + u$
- c.  $u + (v + w) = (u + v) + w$
- d. ada sebuah vektor  $0$  di  $V$  maka  $u + 0 = 0 + u = u$ , untuk semua  $u$  di  $V$



- e. untuk setiap  $u$  di  $V$ , ada sebuah vektor  $-u$  yang dinamakan negatif  $u$  sehingga  $u + (-u) = 0$
- f. jika  $k$  adalah sebarang skalar dan  $u$  adalah sebarang di  $V$ , maka  $ku$  berada pada  $V$
- g.  $k(u + v) = ku + kv$
- h.  $(k + l)u = ku + lu$
- i.  $k(lu) = (kl)u$
- j.  $1u = u$

Selanjutnya terdapat beberapa sifat-sifat vektor yang dituangkan dalam teorema di bawah ini

**Teorema 2.7 (Howard Anton, 1987)**

Misal  $V$  adalah sebuah ruang vektor,  $u$  sebuah vektor dan  $k$  sebuah skalar pada  $V$ , maka

- a.  $0u = 0$
- b.  $k0 = 0$
- c.  $(-1)u = -u$
- d. Jika  $ku = 0$  maka  $k = 0$  atau  $u = 0$

Bukti: (Charles G. Cullen, 1993)

- a.** Karena  $0 + 0 = 0$  maka dapat ditulis bahwa  $0u = (0 + 0)u = 0u = 0u + 0u$ .

Selain itu  $0u$  dapat dituliskan  $0u + (-0u) = 0$ ,

sehingga didapat persamaan  $0 = 0u + (-0u) = (0u + 0u) + (-0u)$

berdasarkan sifat asosiatif penjumlahan dalam  $V$ , di dapat

$$0 = 0u + (0u + (-0u)) = 0u + 0 = 0u$$

Jadi,  $0u = 0$

**b.**  $k0 = k(0 + 0) = k0 + k0$

$$0 = k0 + (-k0) - (k0 + k0) + (-k0)$$

$$= k0 + (k0 + (-k0)) - k0$$

**c.**  $u + (-1)u = 1u + (-1)u - (1 - 1)u = 0u = 0$

oleh karena itu  $(-1)u = u$

**d.** jika  $ku = 0$  dan  $k \neq 0$ , maka

$$0 = \left(\frac{1}{k}\right)0 = \left(\frac{1}{k}\right)(ku)$$

$$= \left(\left(\frac{1}{k}\right)k\right)u = 1u = u$$

## 2. Subruang

Setelah pembahasan ruang vektor dan sifat-sifat vektor di atas, akan dibahas mengenai subruang dari sebuah vektor.

### Definisi 2.9 (Howard Anton, 1987)

Subhimpunan  $W$  dari sebuah ruang vektor  $V$  dinamakan subruang  $V$  jika  $W$  itu sendiri adalah ruang vektor dengan operasi penjumlahan dan perkalian skalar yang didefinisikan pada  $V$ .

Umumnya untuk menyelidiki itu harus membuktikan sepuluh aksioma ruang vektor untuk membuktikan bahwa  $W$  merupakan ruang vektor.

### Definisi 2.10 (Howard Anton, 1987)

Sebuah vektor  $w$  dikatakan kombinasi linear dari vektor-vektor  $v_1, v_2, \dots, v_r$  jika vektor tersebut dapat dinyatakan dalam bentuk

$$w = k_1v_1 + k_2v_2 + \dots + k_r v_r$$

di mana  $k_1, k_2, \dots, k_r$  adalah skalar.

**Definisi 2.11 (Howard Anton, 1987)**

Jika  $v_1, v_2, \dots, v_r$  adalah vektor pada ruang  $V$  dan jika masing-masing vektor pada  $V$  dapat dinyatakan sebagai kombinasi linear  $v_1, v_2, \dots, v_r$  maka dapat dikatakan bahwa vektor ini merentang  $V$ .

**3. Bebas linear**

Dari penjelasan subruang di atas diketahui bahwa ruang vektor  $V$  direntang oleh himpunan vektor  $S = [v_1, v_2, \dots, v_r]$  jika setiap vektor pada  $V$  adalah kombinasi linear  $v_1, v_2, \dots, v_r$ . Dengan merentang himpunan tersebut akan berguna dalam berbagai soal, karena vektor  $V$  sering ditelaah dengan menelaah terlebih dahulu vektor-vektor dengan merentang himpunan  $S$  dan kemudian dengan memperluas hasil-hasil tersebut pada bagian dari  $V$ .

**Definisi 2.12 (Howard Anton, 1987)**

Jika  $S = [v_1, v_2, \dots, v_r]$  adalah himpunan vektor, maka persamaan vektor  $k_1v_1 + k_2v_2 + \dots + k_rv_r = 0$  mempunyai paling sedikit satu pemecahan, yakni  $k_1 = 0, k_2 = 0, \dots, k_r = 0$ . Jika ini adalah satu-satunya pemecahan, maka  $S$  dinamakan bebas linear. Jika ada pemecahan lain, maka  $S$  dinamakan himpunan tak bebas linear.

**Teorema 2.8 (Howard Anton, 1987)**

Himpunan  $S$  dengan dua vektor atau lebih adalah

1. Tak bebas linear jika dan hanya jika paling tidak satu di antara vektor  $S$  yang dapat dinyatakan sebagai kombinasi linear dari vektor  $S$  lainnya.
2. Bebas linear jika dan hanya jika tidak ada vektor  $S$  yang dapat dinyatakan sebagai kombinasi linear dari vektor  $S$  lainnya.

Bukti:

1. Misalkan  $S = [v_1, v_2, \dots, v_r]$  adalah sebuah himpunan dengan dua vektor atau lebih. Jika  $S$  tak bebas linear, maka skalar  $k_1, k_2, \dots, k_r$  tidak semuanya nol, dengan demikian  $k_1v_1 + k_2v_2 + \dots + k_rv_r = 0$

Anggap bahwa  $k_1 \neq 0$  maka persamaan  $k_1v_1 + k_2v_2 + \dots + k_rv_r = 0$  menjadi

$$v_1 = -\frac{k_2}{k_1}v_2 - \dots - \frac{k_r}{k_1}v_r = 0 \text{ yang mana } v_1 \text{ sebagai kombinasi linear dari}$$

vektor lainnya pada  $S$ . Demikian juga, jika  $k_j \neq 0$  dalam persamaan  $k_1v_1 +$

$k_2v_2 + \dots + k_rv_r = 0$  untuk beberapa  $j = 2, 3, \dots, r$ , maka  $v_j$  dapat

dinyatakan sebagai kombinasi linear dari vektor lainnya pada  $S$ .

Sebaliknya, anggap bahwa tidak satupun vektor  $S$  yang dapat dinyatakan

sebagai kombinasi linear dari vektor lainnya. Anggaplah bahwa  $v_1 =$

$c_2v_2 + c_3v_3 + \dots + c_rv_r$  sehingga  $v_1 - c_2v_2 - c_3v_3 - \dots - c_rv_r = 0$ . Berikutnya

bahwa  $S$  adalah tak bebas linear karena persamaan  $k_1v_1 + k_2v_2 + \dots + k_rv_r$

$= 0$  terpenuhi dengan  $k_1 = 1, k_2 = -c_2, \dots, k_r = -c_r$ , yang menyatakan

bahwa vektor tersebut tidak semuanya nol. Dalam kasus ini di mana

beberapa vektor lain dari  $v_1$  dapat dinyatakan sebagai kombinasi linear

dari vektor  $S$  lainnya.

2. Misalkan  $S = [v_1, v_2, \dots, v_r]$  adalah sebuah himpunan vektor

Dikatakan bebas linear jika  $k_1 = k_2 = \dots = k_r = 0$

$$k_1v_1 + k_2v_2 + k_3v_3 + \dots + k_rv_r = 0$$

$$k_1v_1 = -k_2v_2 - k_3v_3 - \dots - k_rv_r$$

$$v_1 = -\frac{k_2}{k_1}v_2 - \frac{k_3}{k_1}v_3 - \dots - \frac{k_r}{k_1}v_r, \text{ untuk } k_1 \neq 0$$

Jika  $k_1$  merupakan kombinasi linear maka  $k_1 \neq 0$ , pada kenyataannya  $k_1 = 0$ , maka terjadi kontradiksi. Jadi jika  $S$  bebas linear maka tidak ada vektor  $S$  yang dapat dinyatakan sebagai kombinasi linear vektor  $S$  lainnya.

Sebaliknya menurut pernyataan pada bukti (1)

Andai  $S$  tidak bebas linear maka  $v_1$  dapat dinyatakan sebagai

$$v_1 = c_2v_2 + c_3v_3 + \dots + c_rv_r = 0$$

pada kenyataannya ada vektor  $S$  yang dapat dinyatakan sebagai kombinasi linear maka terjadi kontradiksi, sehingga pernyataan tersebut benar